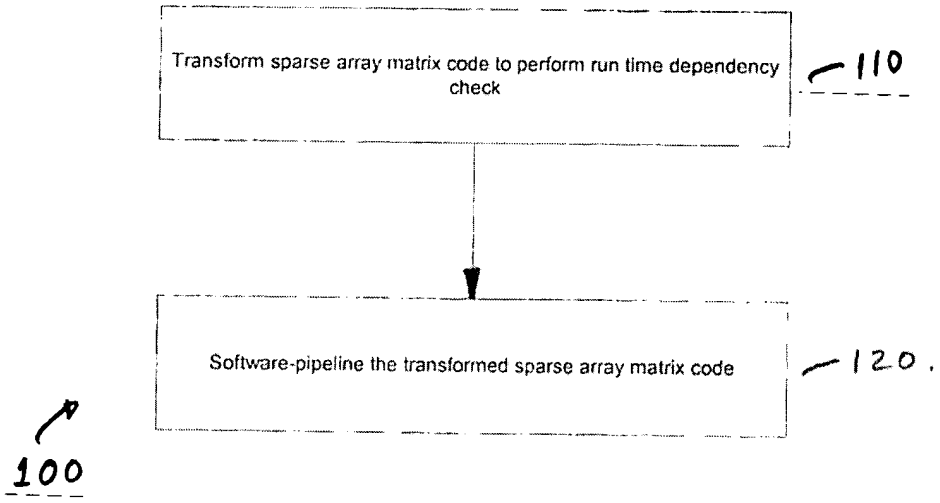

Transform sparse array matrix code to perform run time dependency check
110
Software-pipeline the transformed sparse array matrix code
120
100

FIG. 1

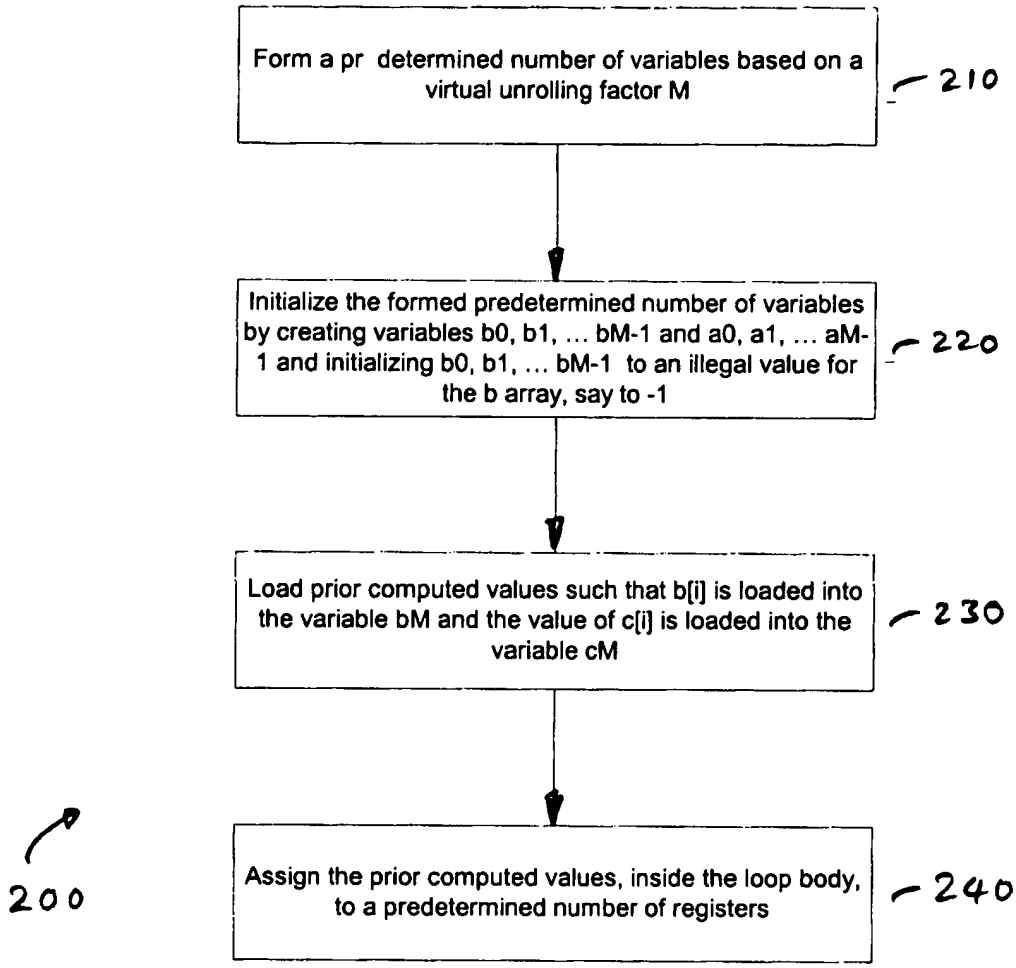

Form a pr determined number of variables based on a virtual unrolling factor M
210
Initialize the formed predetermined number of variables by creating variables b0, b1, ... bM-1 and a0, a1, ... aM-1 and initializing b0, b1, ... bM-1 to an illegal value for the b array, say to -1
220
Load prior computed values such that b[i] is loaded into the variable bM and the value of c[i] is loaded into the variable cM
230
Assign the prior computed values, inside the loop body, to a predetermined number of registers
240
200

FIG. 2

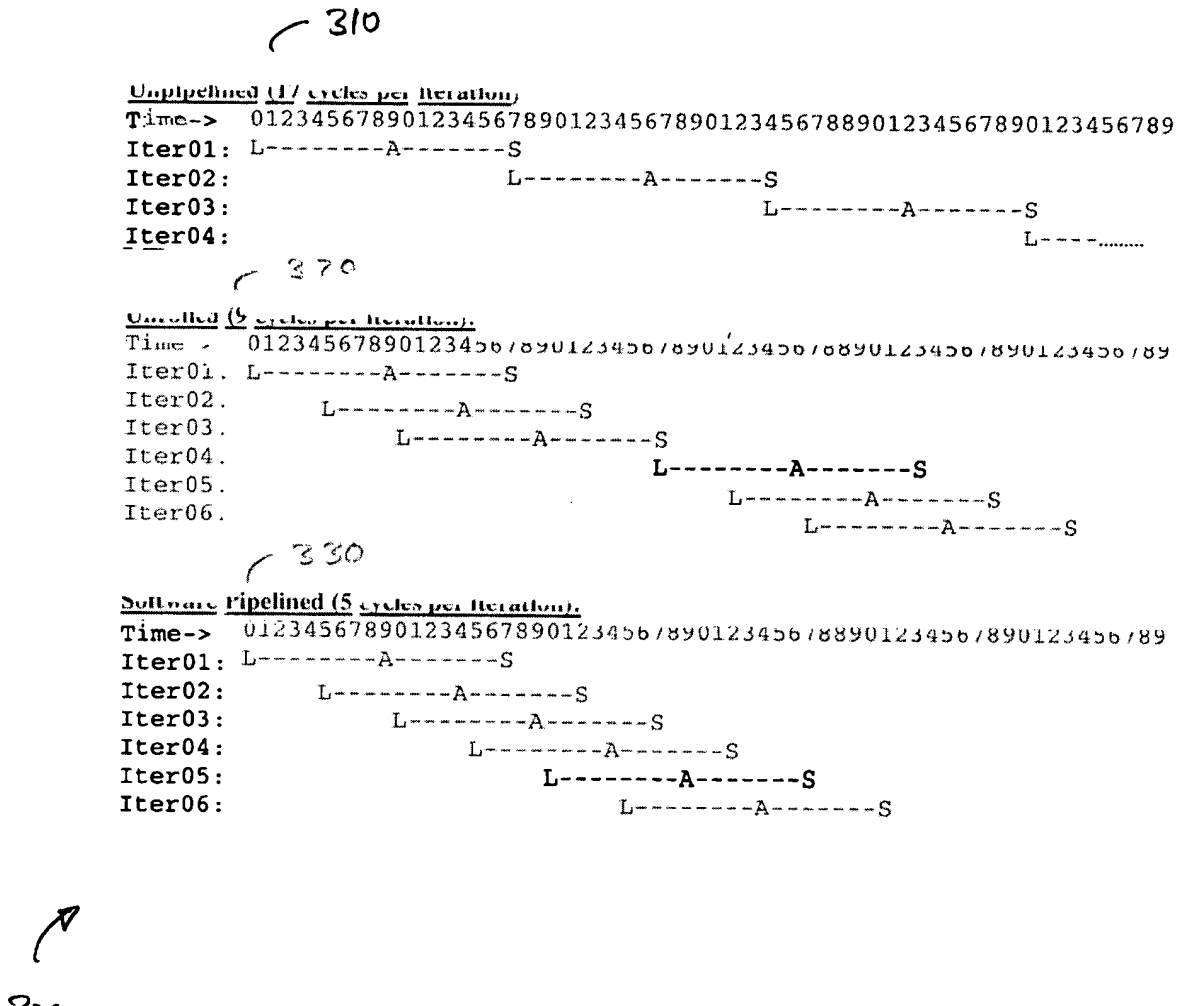
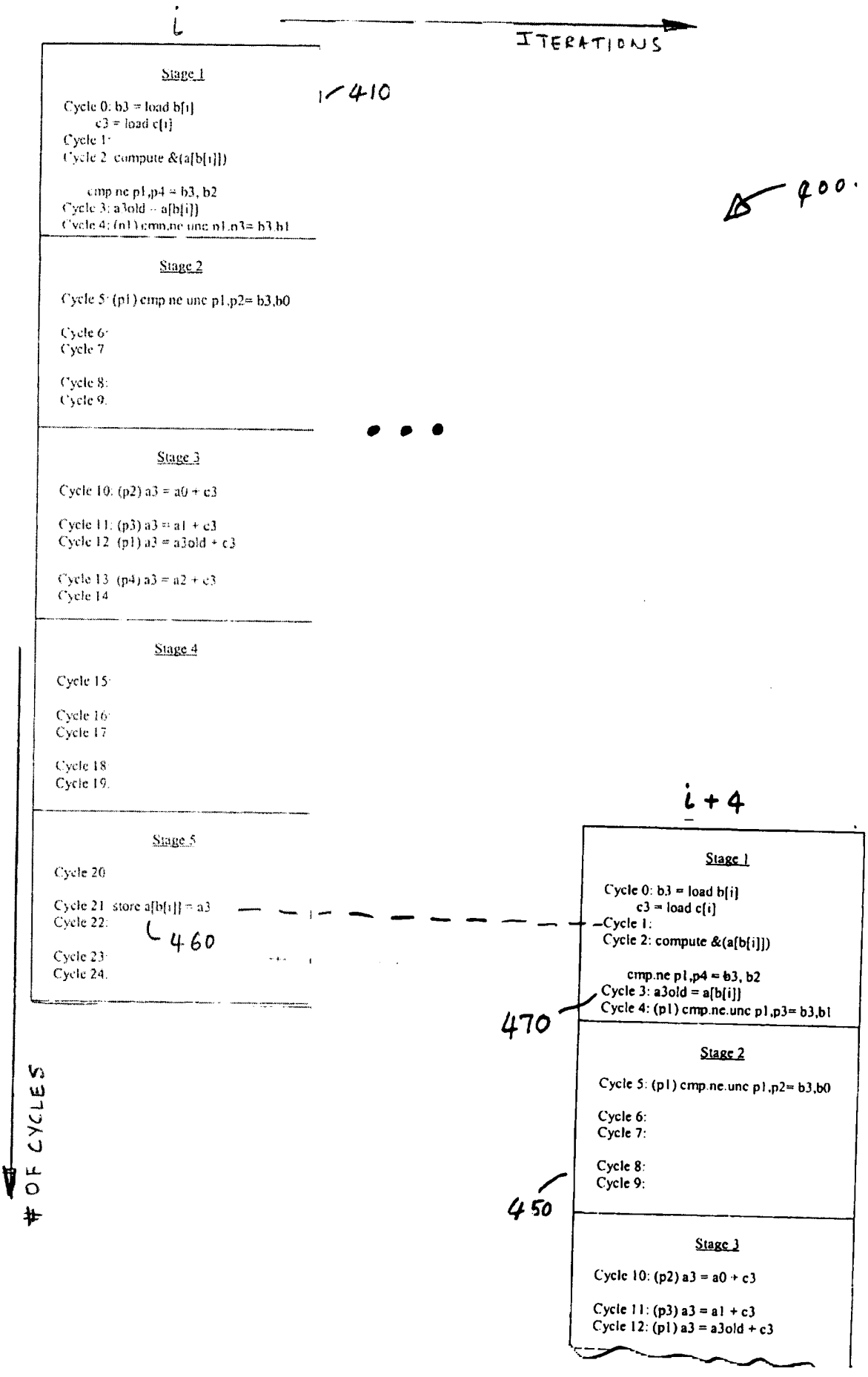

310
Unpipelined (17 cycles per iteration)
Time-> 01234567890123456789012345678901234567889012345678901234567890123456789
Iter01: L--------A-------S
Iter02: L--------A-------S
Iter03: L--------A-------S
Iter04: L----.....
320
Unrolled (5 cycles per iteration).
Time -> 0123456789012345678901234567890123456789012345678901234567890123456789
Iter01. L--------A-------S
Iter02. L--------A-------S
Iter03. L--------A-------S
Iter04. L--------A-------S
Iter05. L--------A-------S
Iter06. L--------A-------S
330
Software ripelined (5 cycles per iteration).
Time-> 0123456789012345678901234567890123456789012345678901234567890123456789
Iter01: L--------A-------S
Iter02: L--------A-------S
Iter03: L--------A-------S
Iter04: L--------A-------S
Iter05: L--------A-------S
Iter06: L--------A-------S
300

FIG 3


i
ITERATIONS
Stage 1
Cycle 0: b3 = load b[i]
c3 = load c[i]
Cycle 1:
Cycle 2: compute &(a[b[i]])
cmp.ne p1,p4 = b3, b2
Cycle 3: a3old = a[b[i]]
Cycle 4: (p1) cmp.ne.unc p1,p3= b3,b1
410
400
Stage 2
Cycle 5: (p1) cmp.ne.unc p1,p2= b3,b0
Cycle 6:
Cycle 7:
Cycle 8:
Cycle 9:
Stage 3
Cycle 10: (p2) a3 = a0 + c3
Cycle 11: (p3) a3 = a1 + c3
Cycle 12: (p1) a3 = a3old + c3
Cycle 13: (p4) a3 = a2 + c3
Cycle 14
Stage 4
Cycle 15:
Cycle 16:
Cycle 17
Cycle 18:
Cycle 19:
Stage 5
Cycle 20
Cycle 21: store a[b[i]] = a3
Cycle 22:
460
Cycle 23:
Cycle 24:
OF CYCLES
i + 4
Stage 1
Cycle 0: b3 = load b[i]
c3 = load c[i]
Cycle 1:
Cycle 2: compute &(a[b[i]])
cmp.ne p1,p4 = b3, b2
Cycle 3: a3old = a[b[i]]
Cycle 4: (p1) cmp.ne.unc p1,p3= b3,b1
470
Stage 2
Cycle 5: (p1) cmp.ne.unc p1,p2= b3,b0
Cycle 6:
Cycle 7:
Cycle 8:
Cycle 9:
450
Stage 3
Cycle 10: (p2) a3 = a0 + c3
Cycle 11: (p3) a3 = a1 + c3
Cycle 12: (p1) a3 = a3old + c3

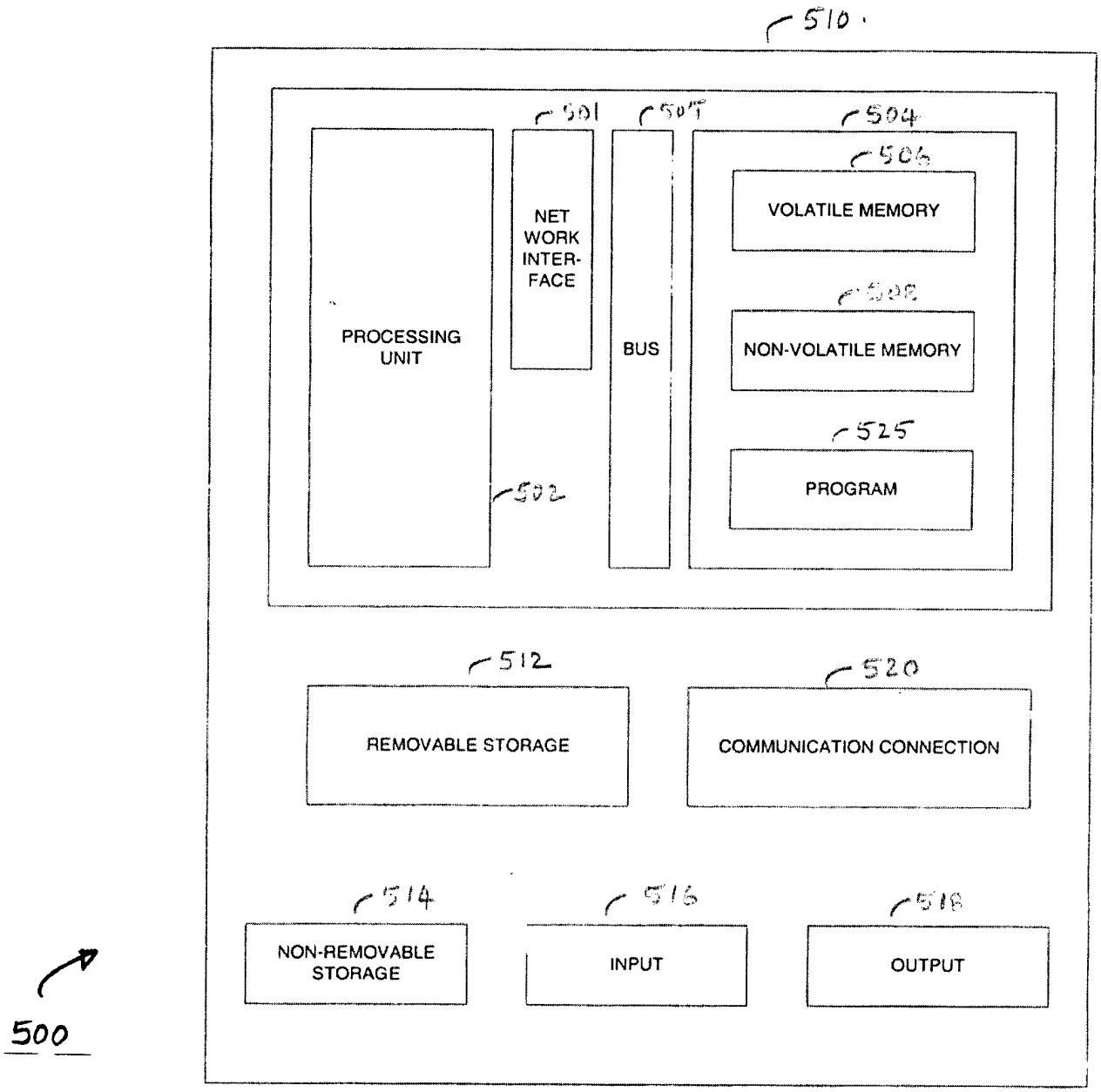

510
501
507
504
506
PROCESSING UNIT
NET WORK INTER-FACE
BUS
VOLATILE MEMORY
508
NON-VOLATILE MEMORY
525
PROGRAM
502
512
520
REMOVABLE STORAGE
COMMUNICATION CONNECTION
514
516
518
NON-REMOVABLE STORAGE
INPUT
OUTPUT
500

FIG. 5